



Joachim
Schlosser

6. Auflage

**Wissenschaftliche Arbeiten
schreiben mit**

L^AT_EX

Leitfaden für Einsteiger

Joachim Schlosser

Wissenschaftliche Arbeiten schreiben mit \LaTeX

Leitfaden für Einsteiger



mitp

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie. Detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Bei der Herstellung des Werkes haben wir uns zukunftsbewusst für umweltverträgliche und wiederverwertbare Materialien entschieden. Der Inhalt ist auf elementar chlorfreiem Papier gedruckt.

ISBN 978-3-95845-290-9

6., überarbeitete Auflage 2016

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2016 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Sabine Schulz

Sprachkorrektur: Marita Böhm, Petra Heubach-Erdmann

Satz: Dr. Joachim Schlosser, www.schlosser.info

Cover: © Steffen Lohse-Koch - fotolia.de

www.latexbuch.de

*Für Julia, weil sie da ist,
und für Quirin und Magali, weil sie so gerne lesen.*

Inhaltsverzeichnis

I Einführung	13
1 Warum \LaTeX?	15
1.1 Inhalt und Layout: Zwei separate Aufgaben	15
1.2 Portabilität und Sicherheit	16
1.3 Seit Langem verfügbar und weiterentwickelt	17
1.4 Warum dieses Buch?	17
1.5 Website zum Buch – www.latexbuch.de	18
2 Aufbau und Installation	19
2.1 Softwarepakete	19
2.1.1 \LaTeX -Distribution: \TeX Live	19
2.1.2 Editor: TeXworks	20
2.1.3 Grafiken in PostScript mit Ghostscript	20
2.2 Installation und Konfiguration	20
2.2.1 \TeX Live herunterladen und installieren	21
2.2.2 Zusätzliche Schriften einrichten	22
2.2.3 Grafikerstellung und -konvertierung unter Windows	22
2.3 Wenn etwas nicht funktioniert	23
3 Erste Schritte in \LaTeX	25
3.1 Das Arbeiten mit \LaTeX	25
3.2 Der Editor	25
3.3 Ein Dokument erstellen und übersetzen	26
3.4 Der Betrachter	28
3.5 Das Dokument	29

II	L^AT_EX-Elemente	33
4	Grundlagen der Textformatierung und Strukturierung	35
4.1	Texteingabe	35
4.2	Grundbefehle	36
4.2.1	Dokumentklasse: Was soll es werden	37
4.2.2	Gliederung in Kapitel und Abschnitte	40
4.2.3	Textauszeichnung mit Schriftschnitten und -größen	43
4.2.4	Logische Textauszeichnung: einfach formatieren . .	46
4.3	Alles, was trennt und verbindet	48
4.3.1	Wortabstände	48
4.3.2	Vertikale Abstände	49
4.3.3	Punkte	50
4.3.4	Binde- und Gedankenstriche	51
4.3.5	Silbentrennung	51
4.3.6	Ligaturen	53
4.4	Aufzählungen, Nummerierungen und Beschreibungslisten . .	54
4.4.1	Standardumgebungen für Listen	54
4.4.2	Listen verändern mit <code>enumitem</code>	55
4.4.3	Andere Aufzählungszeichen	58
4.4.4	Hinweis zur Arbeit mit Listen	59
5	Weitere Befehle und Umgebungen zur Textformatierung und Strukturierung	61
5.1	Besondere Formatierung einzelner Textstücke	61
5.1.1	Flattersatz versus Blocksatz	61
5.1.2	Zitatumgebungen	62
5.1.3	Theoreme	63
5.1.4	Schreibmaschinenausgabe	65
5.1.5	Zusammenfassungen einfügen	66
5.1.6	Eigene Umgebungen	66
5.2	Fußnoten	68
5.2.1	Fußnoten mehrfach verwenden	69
5.2.2	Fußnoten in Überschriften	69
5.3	Querverweise	70
5.3.1	Einfache Querverweise mit L ^A T _E X	70
5.3.2	Querverweise mit Seitenreferenz: <code>Varioref</code>	72
5.3.3	Querverweise mit automatischem Typ: <code>cleveref</code> . . .	74
5.4	Maßeinheiten	75
5.4.1	Zahlen, Exponenten, Winkel	76
5.4.2	Einheitenbehaftete Zahlen	78

5.5	Quellcode setzen mit listings	79
5.5.1	Aussehen von Quellcode anpassen	82
5.5.2	Zeilennummern anzeigen und Zeilen auswählen . . .	84
5.5.3	Eigene Quellcodeumgebungen definieren	86
5.5.4	Quellcode als Gleitobjekt und Ausblick	86
6	Seitenaufbau	89
6.1	Satzspiegel	89
6.1.1	Satzspiegel, was ist das?	89
6.1.2	Satzspiegel mit KOMA-Script	90
6.1.3	Zeilenabstand	93
6.1.4	Satzspiegel frei anpassen	93
6.2	Layout	95
6.2.1	Layoutparameter	95
6.2.2	Kopf- und Fußzeilen sowie Seitenzahlen	97
6.2.3	Eigene Kopf- und Fußzeilen	99
6.3	Gleitobjekte	102
6.3.1	Die figure-Umgebung	103
6.3.2	Die table-Umgebung	105
6.4	Titelei	105
6.4.1	Grundelemente des Titels	105
6.4.2	Erweiterte Titelei	106
7	Tabellen	111
7.1	Tabellengrundlagen	112
7.1.1	Was sind Tabellen?	112
7.1.2	Tabellen definieren	112
7.1.3	Mehrspaltige und mehrzeilige Zellen	114
7.2	Flexible Tabellen	116
7.2.1	Präfixe und Suffixe in Spalten, Spaltenausrichtung .	116
7.2.2	Ausrichten von Spalten an Dezimalzeichen	118
7.2.3	Tabellen mit fester Gesamtbreite	121
7.2.4	Lange Tabellen mit mehr als einer Seite	122
7.3	Aussehen von Tabellen	126
7.3.1	Schlicht und elegant: booktabs	127
7.3.2	Farbige Tabellen	130
7.3.3	Tabellen quer drucken	131
8	Grafiken und Bilder	135
8.1	Grafiken einfügen	135
8.1.1	Dateiformate	135

8.1.2	Abbildungen erstellen	136
8.1.3	Die wichtigsten Optionen für die Grafikeinbindung .	137
8.2	Grafiken erzeugen	140
8.2.1	Vektor versus Bitmap	141
8.2.2	Vektorgrafiken erstellen: TikZ/pgf	142
8.2.3	Vektorgrafiken mit picture, PSTricks, MetaPost . .	158
8.2.4	Bitmaps erstellen und umwandeln	164
8.2.5	Externe Grafikanwendungen	165
8.3	Grafiken nachträglich beschriften: overpic	167
8.4	Untergrafiken mit Subcaption	169
8.5	Textumflossene Grafiken	170
9	Literatur zitieren und verwalten mit biblatex	171
9.1	Literaturzitate verwenden	171
9.1.1	Ein Beispiel	172
9.1.2	Aufbau der BibTeX-Datei	174
9.1.3	BibTeX-Eintragstypen	174
9.1.4	BibTeX-Feldtypen	178
9.1.5	Mehrere Autoren, Herausgeber oder Organisationen	183
9.1.6	Weitere Eigenschaften von BibTeX-Dateien	183
9.2	Zitieren und Erzeugen des Literaturverzeichnisses	184
9.2.1	Zitieren	184
9.2.2	Erzeugen des Literaturverzeichnisses	186
9.3	Zitierstile jeder Art mit biblatex	187
9.3.1	Numerische Stile	187
9.3.2	Mit Autor und Jahr zitieren	188
9.3.3	Mit Autor und Titel zitieren	190
9.3.4	Mit Kürzel zitieren	191
9.3.5	Den ganzen Verzeichniseintrag zitieren	192
9.4	Das Literaturverzeichnis in biblatex formatieren	194
9.4.1	Sortieren der Einträge	194
9.4.2	Titel und Anmerkungen des Literaturverzeichnisses	195
9.4.3	Unterteilte Literaturverzeichnisse	200
9.5	Sigel – Shorthands	205
9.6	Ausblick	206
10	Verzeichnisse aller Art	207
10.1	Inhaltsverzeichnis	207
10.1.1	Gliederungsebenen einstellen	208
10.1.2	Einträge manuell hinzufügen	208
10.1.3	Aussehen des Inhaltsverzeichnisses ändern	210

10.2	Abbildungs- und Tabellenverzeichnis	216
10.3	Index erstellen	217
10.3.1	Durchstich – flott zum Index	219
10.3.2	Auszeichnung der Begriffe im Dokument	220
10.3.3	Xindy & TeXindy: Wortliste verarbeiten, Index erzeugen	222
10.3.4	Aussehen des Index ändern	223
10.3.5	Erweiterte Formatierung des Index	226
10.4	Glossar und Abkürzungsverzeichnis erstellen	229
10.4.1	Glossar im Eigenbau	230
10.4.2	Durchstich – der kürzeste Weg mit Glossaries	230
10.4.3	Glossareinträge erstellen	231
10.4.4	Glossareinträge im Text markieren	233
10.4.5	Das Glossar erzeugen und Aussehen anpassen	233
10.4.6	Miteinander verbundene Glossareinträge erstellen	237
11	Mathematische Ausdrücke	239
11.1	Mathematikmodus und Gleichungseingabe	239
11.1.1	Einfache mathematische Ausdrücke	240
11.2	Mathematikumgebungen	242
11.2.1	Übergeordnete Gleichungsumgebungen	242
11.2.2	Untergeordnete Gleichungsumgebungen	245
11.3	Abstände, Klammern und Punkte	246
11.4	Die ganze Welt der Mathematik	247
11.4.1	Matrizen	247
11.4.2	Drüber und drunter: Pfeile, Brüche, Operatoren	248
11.4.3	Funktionen	250
11.4.4	Chemische Gleichungen und andere Symbole	250
11.5	Schriften in Gleichungen	250
11.6	Tabellen der Symbole und Befehle für Gleichungen	251
12	Längere Arbeiten aufsetzen	259
12.1	Mehrere Dateien verwenden	259
12.1.1	Befehle zum Teilen von Dokumenten	259
12.1.2	Hinweise zum Untergliedern	261
12.2	Versionierung	261
12.2.1	Git als Software für die Versionierung	262
12.2.2	Versionierung und L ^A T _E X	263

III Fertigstellen der Arbeit	265
13 Schriften und Schriftfamilien	267
13.1 Bemerkung	267
13.2 Grundlagen	268
13.2.1 Pixel- und Vektorschriften	268
13.2.2 Grundlagen der Schriftenwahl	268
13.2.3 Serifen oder serifenlos?	269
13.3 Schriftpakete	270
13.3.1 Computer Modern	271
13.3.2 Latin Modern	272
13.3.3 T _E X Gyre	273
13.3.4 Nicht-freie Schriften	277
13.3.5 Weitere Schriften	279
13.4 Optischer Randausgleich	280
14 Spezialitäten bei PDF	283
14.1 Der Weg zum PDF	283
14.2 Aktive Querverweise mit hyperref	284
14.2.1 Links kenntlich machen	284
14.2.2 Lesezeichen	285
14.2.3 Meta-Angaben	286
14.2.4 Textverändernde Optionen	286
14.2.5 Zusätzliche Verweisbefehle für den Text	287
14.3 Schnittmarken: Drucken von kleineren Seiten	288
15 Fehlersuche	291
15.1 Häufige Fehlerquellen	291
15.2 Beliebte Fehlermeldungen und Warnungen	291
16 Ausblick	301
16.1 Benutzen bereits bestehender Vorlagen	301
16.2 Weiterführende Hilfe	301
16.3 Ausblick, Dank und Aufforderung	303
Über den Autor	305
Liste der vorgestellten Pakete	307
Literatur	309
Stichwortverzeichnis	315

Teil I

Einführung

Warum L^AT_EX?

1.1 Inhalt und Layout: Zwei separate Aufgaben

Wenn Sie dieses Buch in Händen halten, haben Sie Ihre Entscheidung wahrscheinlich bereits getroffen: Sie wollen eine wissenschaftliche Arbeit schreiben, und zwar nicht mithilfe gängiger Textverarbeitungsprogramme, sondern mit L^AT_EX. Warum ist das eine gute Entscheidung? Einige Fakten:

L^AT_EX (ausgesprochen »Lah-Tech«) ist anders als Textverarbeitungsprogramme. Ganz anders. In einem Textverarbeitungsprogramm wie WordTM schreiben Sie Ihren Text und bestimmen gleichzeitig das Aussehen Ihres Dokuments. Da Sie jede Änderung des Dokuments sofort sehen können, spricht man hier gemeinhin von *What you see is what you get* (WYSIWYG).

In L^AT_EX sehen Sie hingegen die Ausgabe während des Schreibens nicht immer vor sich. Die Gestaltung der Textabschnitte steht also zunächst einmal nicht im Vordergrund, sondern es wird der reine Text geschrieben und zusätzlich ein sogenanntes *Markup*. Das bedeutet, man fügt spezielle Layoutbefehle ein. Der Gestalter – L^AT_EX – nimmt dieses *Markup* und interpretiert es, stellt das Layout zusammen und legt fest, wie die einzelnen Textteile auszusehen haben. Der Setzer schließlich – T_EX – erstellt die druckfertige Ausgabe aus den Anweisungen des Gestalters.

Das hat natürlich auch ein paar wenige Nachteile. Zum einen ist der größte zweifelsohne der, dass man die Layoutbefehle kennen muss – wobei man sich von sogenannten L^AT_EX-Editoren unterstützen lassen kann. Zum anderen kann das Dokument gerade bei umfangreichen Texten unter Umständen während der Eingabe unübersichtlicher werden, da bei vielen Editoren die Tabellen, Abbildungen und Formeln nicht sofort angezeigt werden. Warum geht das nicht in einem Aufwasch wie bei Word? Warum ist es die Mühe wert?

An ein Dokument, insbesondere an eine wissenschaftliche Arbeit, wird nicht nur ein inhaltlicher Anspruch gestellt. Es geht auch darum, dass das Lesen ermüdungsfrei und angenehm gestaltet wird. Das Layout des Dokuments hat also nicht besonders auffällig oder künstlerisch extravagant zu sein, sondern in erster Linie diesen Zweck zu erfüllen. Nicht umsonst

war und ist die Gestaltung eines Schriftstücks als auch dessen Satz eine Handwerkskunst, die seit Jahrhunderten gelehrt wird.

Seit es Textverarbeitungsprogramme gibt, ist diese Trennung der Aufgaben nicht mehr gewährleistet. L^AT_EX stellt die ursprüngliche Trennung der Aufgaben wieder her. Während des Schreibens kümmern Sie sich zunächst nicht darum, wie etwas später *aussehen* soll, sondern konzentrieren sich allein auf Inhalt und *Bedeutung* des Textabschnitts. Sie werden sehen, wie dies an Qualität gewinnt, sobald der Kopf frei dafür ist, eben *weil* man nicht immer das endgültige Aussehen vor sich hat. Seien Sie ehrlich: Wie oft mussten Sie bis jetzt die inhaltliche Arbeit an einem Text unterbrechen, weil Sie zum Beispiel am Layout einer Tabelle feilten? L^AT_EX trennt den Inhalt vom Layout, stellt aber die Option zur Verfügung, jederzeit eine Voransicht zu erzeugen, die Sie am Bildschirm begutachten oder drucken können.

1.2 Portabilität und Sicherheit

Textverarbeitungsprogramme speichern ihre Dokumente in sogenannten *Binärdateien*, also Dateien, die man nur mit dem Programm lesen kann, mit dem sie auch gespeichert wurden. Ist die Datei defekt, weil beispielsweise durch einen Absturz eines Programms oder des Betriebssystems Teile durcheinandergekommen sind oder fehlen, ist das Dokument in der Regel nicht zu retten.

L^AT_EX wird in simplen Textdateien gespeichert, aus denen später die Ausgabe erzeugt wird. Diese Textdateien können mit jedem beliebigen Editor bearbeitet werden. Selbst wenn ein partieller Datenverlust entsteht oder Teile durcheinanderkommen, ist der Rest auf jeden Fall weiterhin lesbar. Die Gefahr eines totalen Datenverlusts geht gegen null. Dieses Speichern in einfachen Textdateien sorgt auch dafür, dass L^AT_EX-Dokumente unter jedem Betriebssystem bearbeitet werden können, selbst wenn kein L^AT_EX-System installiert oder verfügbar sein sollte. Aus dem gleichen Grund sind mit L^AT_EX gesetzte Dokumente auch immer *reproduzierbar*. Selbst Jahre später können alte Dokumente immer noch gelesen und verarbeitet werden.

Schlussendlich hat dies auch Auswirkungen auf den Umgang mit sehr großen Dokumenten: Selbst das Bearbeiten von umfangreichen Dokumenten mit mehreren tausend Seiten ist kein Hindernis unter L^AT_EX. Diese Zuverlässigkeit ist ebenfalls ein Ergebnis der Trennung von Inhalt und Layout.

1.3 Seit Langem verfügbar und weiterentwickelt

Seit nunmehr über fünfunddreißig Jahren ist das Satzprogramm $\text{T}_{\text{E}}\text{X}$ ein stabiles Programm, bei dem schon seit Langem keine Fehler mehr gefunden wurden. Donald E. Knuth hat es seinerzeit für sein Buch »The Art of Computer Programming« entwickelt, weil es kein vernünftiges System für den rechnergestützten Satz von Büchern und mathematischen Formeln gab. Da $\text{T}_{\text{E}}\text{X}$ aber sehr schwierig zu benutzen ist, wurde vor mehr als fünf- undzwanzig Jahren der Aufsatz für die Gestaltung – $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ – entwickelt, der es wesentlich einfacher machte, das Satzprogramm anzuwenden. Hierfür wiederum entstanden seitdem viele Erweiterungen für fast alle Arten von Dokumenten. Fortlaufend erfährt das Programm Verbesserungen und Veränderungen aus aller Welt. Leistungsfähige Pakete und Programme erlauben das Erzeugen von PDF inklusive Verlinkungen, Zitierstilen aller Art, speziellen Formatierungen und so weiter.

Die Weiterentwicklungen werden auch von Vereinen gefördert: den *T_EX User Groups*. Die deutsche TUG heißt Dante e. V., *Deutsche Anwendervereinigung T_EX*. Werden Sie Mitglied oder spenden Sie an Dante e. V., wenn Sie etwas an die Entwicklung zurückgeben wollen.

1.4 Warum dieses Buch?

Dieses Buch möchte Ihnen helfen, die Anfangshürden zu überwinden, damit Sie rasch zu vollständigen, druckfähigen Dokumenten gelangen. Wir halten uns deswegen nicht lange mit Theorien auf, sondern wagen den direkten Einstieg. Durch zahlreiche Praxisbeispiele werden Sie in der Lage sein, nachzuvollziehen, wie das Programm arbeitet. Das erklärte Ziel ist es, Sie bereits während der Lektüre dieses Buches in die Lage zu versetzen, eine wissenschaftliche Arbeit mit $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ zu erstellen, sei es eine Fach-, Studien-, Diplom-, Bachelor- oder Masterarbeit oder gar eine Dissertation. Andere Dokumentarten wie Briefe werden zwar angesprochen, jedoch nicht weiter ausgeführt.

Dieses Buch will kein allumfassendes Werk zu $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ sein: Der Fokus liegt ganz klar auf der *Nutzung*, was für die allermeisten Aufgaben ausreicht. Deshalb wird nicht erklärt, wie Sie eigene *Klassen* und *Styles* selbst schreiben, sondern lediglich behandelt, wie Sie vorhandene verwenden. Die *Erweiterung* oder *Entwicklung* ist also ausgeklammert.

Anders als viele andere Einsteigerwerke zu $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ erläutert dieses Buch einfachere Zusatzpakete in der Regel gleich beim entsprechenden Thema. Auf diese Weise haben Sie die Informationen an *einer* Stelle zum Nachlesen.

Und damit auch garantiert nichts schiefgeht, wird natürlich auch die Installation eines L^AT_EX-Systems sowie das Arbeiten mit Literatur und Grafiken beschrieben. Der Umfang, in dem auf die einzelnen Themen eingegangen wird, ist auf das Erstellen einer größeren wissenschaftlichen Arbeit abgestimmt. Sollten darüber hinaus spezielle Fragen auftauchen, so verweise ich am Ende des Buches auf Hilfen, die Sie über dieses Buch hinaus nutzen können.

1.5 Website zum Buch – www.latexbuch.de

Zu diesem Buch gehört die Website www.latexbuch.de. Dort finden Sie neben den Codebeispielen zu allen Kapiteln auch eine aktuelle Installationsanleitung mit regelmäßig aktualisierten Quellen für die Programmpakete. Eventuelle Fehlerkorrekturen finden Sie ebenfalls dort.

Tipps und Tricks zu L^AT_EX und dem Schreiben von Texten können Sie in meinem Blog auf derselben Seite lesen.

Bitte laden Sie sich die Beispiele von www.latexbuch.de/beispiele herunter.

Aufbau und Installation

Damit Sie auch gleich das Gelernte ausprobieren können, sollten Sie ein wenig Zeit investieren und der Installationsanleitung folgen, damit Sie das \LaTeX -System auf Ihrem Rechner haben.

Die vollständige Anleitung mit aktuellen Quellen dazu finden Sie auf der Website zu diesem Buch: <http://www.latexbuch.de/installieren>. Hier wird die Installation nur sehr verkürzt besprochen. Bitte lesen Sie die Onlinefassung, falls Ihnen das hier zu schnell geht.

Alle in diesem Kapitel genannten Programme und Pakete sind mindestens kostenlos, meist sogar freie Software¹.

Trotz der prinzipiellen Kostenfreiheit lege ich Ihnen ans Herz, sich bei den einzelnen Projekten bei Gefallen auch finanziell erkenntlich zu zeigen, Einzelheiten können Sie auf den Websites der Projekte nachlesen. Wenn Sie zum Beispiel Mitglied bei der Deutschsprachigen Anwendervereinigung \TeX – dante e. V. – werden, dann bekommen Sie einmal im Jahr automatisch die neueste \TeX Live DVD zugesandt.

2.1 Softwarepakete

Um mit \LaTeX arbeiten zu können, benötigen Sie neben der eigentlichen \TeX -Distribution weitere Komponenten.

2.1.1 \LaTeX -Distribution: \TeX Live

Die Distribution ist dafür verantwortlich, aus dem Eingabetext das fertige Dokument zu generieren. Im Editor *schreiben* Sie Ihren Text.

Die \LaTeX -Distribution ist der Kern des Systems. Sie beinhaltet die Programme zum Erzeugen von DVI, PostScript und PDF, außerdem alle Zusatzpakete, die erweiterte Möglichkeiten wie z. B. Musiksatz, Standardlayouts für verschiedene Organisationen, Layoutbesonderheiten u. Ä. bieten.

¹Was bekanntlich ein Unterschied zu *kostenlos* ist: »Free as free speech, not free beer.«
Siehe dazu die Seiten der *Free Software Foundation* unter <http://www.fsf.org>.

Für dieses Buch verwenden wir TEX Live. Diese Distribution ist meines Erachtens nach gut zu installieren, funktioniert prächtig und wird gut gepflegt. Es sind ca. 2 Gigabyte herunterzuladen.

2.1.2 Editor: TeXworks

Die Distribution ist nur dafür verantwortlich, aus der Eingabe die Ausgabe zu generieren. Die Eingabe selbst nehmen Sie im Editor Ihrer Wahl vor. Die Wahl fällt für dieses Buch auf TeXworks, der bei der TEX Live schon dabei ist.

Die Erweiterung TeXlipse² in Eclipse bietet eine Umgebung für $\text{L}\text{A}\text{T}\text{E}\text{X}$ für alle, die ohnehin in Studium oder Beruf mit Eclipse programmieren. Auf die Installation gehe ich nicht ein.

TeXworks ist für mausorientierte Benutzer gut geeignet. Er bietet neben grafischen Menüs für die meisten mathematischen Symbole und anderen Befehle auch während des Tippens von Befehlen mögliche Vervollständigungen als sogenannte *Tooltips*. Gerade für den Neuling eignet sich TeXworks, auch weil Sie nichts extra installieren müssen. Später können Sie dann auf das »Profifgerät« Emacs umsteigen.

2.1.3 Grafiken in PostScript mit Ghostscript

PostScript ist in der Unix-Welt *das* Austauschformat für Vektorgrafiken. So ist es auch unter $\text{L}\text{A}\text{T}\text{E}\text{X}$ am besten, Grafiken als (Encapsulated) PostScript (PS/EPS) einzubinden.

Aus allen Windows-Anwendungen wie z. B. Word oder PowerPoint oder aus Bitmaps lässt sich PostScript erzeugen, selbst wenn die jeweilige Anwendung dies nicht direkt unterstützt, indem Sie über einen PostScript-Druckertreiber in eine Datei drucken und in Ghostscript weiterverarbeiten.

2.2 Installation und Konfiguration

Mein System läuft auf Windows 10 und auf Mac OS X. Einige Zugriffspfade zu bestimmten Systemeinstellungen können auf Ihrem System geringfügig von den beschriebenen abweichen.

Zugriffspfade bezeichnen Startmenüeinträge, Schaltflächen und Menüeinträge mit den entsprechenden Namen.

Sie sollten die Reihenfolge der einzelnen Schritte wegen Programmabhängigkeiten unbedingt einhalten. Verwenden Sie ein anderes Verzeichnis

²Taavi Hupponen et al. *TeXlipse*. 2010. <http://texlipse.sourceforge.net>.

für ein Programm, müssen Sie dies in den nachfolgenden Schritten selbst berücksichtigen.

2.2.1 T_EX Live herunterladen und installieren

Laden Sie sich zunächst von <http://www.tug.org/texlive/> den Installer von T_EX Live für Windows oder Linux oder von <http://www.tug.org/mactex/> für Mac OS X. Wenn Sie nur eine dünne Internetverbindung haben, dann bestellen Sie sich bitte die *T_EX Collection DVD* von der Fachbuchhandlung Lehmanns, <http://www.lob.de>.

Windows

Entpacken Sie die Zip-Datei und rufen Sie im entstandenen Verzeichnis das Programm `install-tl.bat` auf. Akzeptieren Sie alle Vorgaben und lassen Sie Ihren Computer für die nächsten Stunden – je nach Internetverbindung – in Ruhe.

Linux

Entpacken Sie die `.tar.gz`-Datei und rufen Sie im entstandenen Verzeichnis das Programm `install-tl` auf, am besten mit Benutzeroberfläche. Öffnen Sie eine Kommandozeile und geben Sie folgende Befehle ein:³

```
cd /pfad/zum/installer
perl install-tl -gui
```

In der erscheinenden Benutzeroberfläche müssen Sie eine Änderung vornehmen: Ziemlich weit unten wählen Sie für »Create symlinks in system directories« bitte »yes«. Auf diese Weise brauchen Sie später nicht extra die Programme zum Suchpfad hinzuzufügen.

Akzeptieren Sie alle anderen Vorgaben und lassen Sie Ihren Computer für die nächsten Stunden – je nach Internetverbindung – in Ruhe.

Mac OS X

Der Installer für Mac OS X enthält bereits alles, was Sie benötigen. Rufen Sie das `MacTeX.pkg` auf und akzeptieren Sie alle Vorgaben.

Für alle Betriebssysteme gilt: Alle Programme können Sie dann direkt von der Kommandozeile aufrufen.

³Berry und Ziegenhagen, *T_EX Live Dokumentation*.

2.2.2 Zusätzliche Schriften einrichten

TEX Live bringt im Paket nur Schriften mit, die von den Lizenzbedingungen auch eine Verteilung auf DVDs erlauben. Somit fehlen einige, bei denen dies nicht gestattet ist, die jedoch trotzdem kostenfrei verwendet werden dürfen. Genau diese installieren Sie jetzt noch nach.

Laden Sie sich dafür `getnonfreefonts`⁴ herunter. Öffnen Sie eine Kommandozeile unter welchem Betriebssystem auch immer und wechseln Sie in das Downloadverzeichnis. Geben Sie folgende zwei Kommandos ein:

```
texlua install-getnonfreefonts  
getnonfreefonts-sys -a
```

Dies installiert zunächst `getnonfreefonts`, bezieht und richtet dann alle verfügbaren Schriften systemweit ein. Sollten Sie keine Administratorrechte haben, dann ersetzen Sie die zweite Zeile:

```
getnonfreefonts -a
```

Auf das Verwenden von Schriften geht Kapitel 13 ein.

2.2.3 Grafikerstellung und -konvertierung unter Windows

Den PostScript-Druckertreiber richten Sie unter Windows ein mit der Maus: START → GERÄTE UND DRUCKER → DRUCKER HINZUFÜGEN. Als Erstes *lokaler Drucker* wählen und als Anschluss den FILE-Port, der richtige Treiber ist der *Apple Color LaserWriter 12/600* (bis Windows Vista) bzw. der *MS Publisher Color Printer* (ab Windows 7, in der Herstellerliste »Generic«). Den Druckernamen auf »PostScript File« setzen.

Nach der Installation navigieren Sie mit der Maus nach START → GERÄTE UND DRUCKER → POSTSCRIPT FILE → DRUCKER → DRUCKEINSTELLUNGEN → ERWEITERT → DOKUMENTOPTIONEN → POSTSCRIPT-OPTIONEN, um folgende Einstellungen vorzunehmen: Die Felder *PostScript-Ausgabeoption* auf »Optimale Portabilität« und *TrueType-Downloadoption* auf »Umriss« setzen. Deaktivieren Sie das *ICM Color Matching*, um Farbverfälschungen zu vermeiden.

Ab sofort kann jede Windows-Anwendung PostScript-Dateien erzeugen, indem Sie den neuen Drucker verwenden. Den erzeugten Dateien sollten Sie immer die Erweiterung `.ps` anstatt des vorgegebenen `.prn` geben.

Es ist überaus sinnvoll, Ghostscript dem Suchpfad hinzuzufügen. Also wieder in START → EINSTELLUNGEN → SYSTEMSTEUERUNG → SYSTEM → ERWEITERT → UMGEBUNGSVARIABLEN der Variablen *PATH* das Verzeichnis,

⁴<http://www.tug.org/fonts/getnonfreefonts/install-getnonfreefonts>

C:\texlive\2013\tlpkg\tlgs\bin\, durch Semikolon abgetrennt hinzufügen.
Wichtig: keine Leerzeichen vor und nach Einträgen in der *PATH*-Variablen!

Um sich auf die Dauer die Arbeit zu erleichtern, empfehle ich, einen »EPS-Drucker« einzurichten, auch dafür gibt es eine Anleitung im Internet⁵.

2.3 Wenn etwas nicht funktioniert

Sollte irgendetwas nicht funktionieren oder sich offensichtlich seltsam verhalten, ist das kein Beinbruch. Häufig liegt es an vergessenen Umgebungsvariablen. So sollte zunächst überprüft werden, ob alle von dieser Anleitung geforderten Umgebungsvariablen entsprechend gesetzt wurden. Außerdem eine beliebte Fehlerquelle: Dateien am falschen Ort. Einfach noch mal nach Anleitung kontrollieren und ggf. korrigieren. Gegebenenfalls sollten Sie dann auch einen Blick in die Installationsanweisung des jeweiligen Pakets werfen, ob dort noch weitere Hinweise zu finden sind.

Die vollständige Anleitung mit aktuellen Quellen dazu finden Sie auf der Website zu diesem Buch: <http://www.latexbuch.de/installieren>.

⁵Joachim Schlosser. *EPS Writer für Windows*. 2005. <http://schlosser.info/epswriter>.

Erste Schritte in \LaTeX

3.1 Das Arbeiten mit \LaTeX

Wie eingangs bereits beschrieben, ist die Arbeitsweise mit \LaTeX eine andere als mit Textverarbeitungsprogrammen. Letztere zwingen den Benutzer, Layout und Aussehen der einzelnen Teile gleich interaktiv bei der Eingabe festzulegen. Anders bei \LaTeX : Hier verfassen Sie den reinen Text. Bezüglich des Layouts bekundet man zunächst seine Absichten mithilfe von »Befehlen«. Das können einzelne Befehle sein oder ganze sogenannte Pakete. Damit können Sie zum Beispiel das Layout von Briefen, Artikeln oder sonstiger Dokumente festlegen. Die Befehle können vollkommen unabhängig vom Erstellen des Textes benutzt werden. Erst nach dem Erfassen des Textes liest der *Übersetzer* diese Befehle und setzt sie in Layout um. Der Autor muss sich also nur um Inhalt und Struktur des Dokuments kümmern, das Aussehen wird von \LaTeX erzeugt. (Selbstverständlich haben Sie die Möglichkeit, das Aussehen ebenfalls über Befehle zu beeinflussen.) Sie öffnen also nicht etwa ein Programm namens \LaTeX und schreiben dort Ihren Text (obwohl Sie auch das tun könnten), sondern arbeiten mit mehreren Programmen. Zum Schreiben verwenden Sie dabei einen separaten Editor, den Sie frei wählen können, je nachdem, mit welchem Sie am besten zurechtkommen. Im *Editor* verfassen Sie Ihr Dokument und geben die Befehle ein. Ein anderes Programm, der \LaTeX -Prozessor, übersetzt die Eingabe dann in das gewünschte Ausgabeformat, beispielsweise PDF.

3.2 Der Editor

Der *Editor* ist das Programm, in dem Sie Ihren Text schreiben und die Befehle eingeben. Editoren mit \LaTeX -Unterstützung helfen bei der Auswahl der Befehle, beim Erstellen des Dokuments und beim Zusammenspielen der einzelnen Dateien. Beispiele dafür sind Emacs¹, Kile, WinEdt oder das in diesem Buch für Einsteiger empfohlene TeXworks.

¹Die Installation von Emacs unter Windows finden Sie im Tutorial Joachim Schlosser. *LaTeX – ein komplettes System unter Windows*. 2016. <http://schlosser.info/latexsystem>.

Ebenfalls in der Verantwortung des Editors liegt – sofern gewünscht – die Rechtschreibprüfung.

Selbst wenn Sie schon eine Weile mit L^AT_EX arbeiten, sollten Sie sich nicht scheuen, den Editor zu wechseln, wenn Sie einen besseren finden. »Besser« ist dabei natürlich von den einzelnen Bedürfnissen abhängig. Der eine arbeitet lieber mit der Maus und entsprechenden Buttons, der andere hat lieber Tastaturkürzel und eine integrierte Vorschau. Für jeden Geschmack existiert ein passender Editor!

In diesem Buch gehe ich nur ein Stück weit auf den Editor TeXworks ein, der für Windows verfügbar ist und für den Einsteiger viele Hilfen bietet. Für GNU/Linux sei an dieser Stelle Kile empfohlen, der ungefähr in derselben Liga spielt.

Um Ihr erstes L^AT_EX-Dokument erstellen zu können, öffnen Sie also zunächst einmal Ihren Editor. Sofern Sie TeXworks verwenden, sehen Sie jetzt in etwa Abbildung 3.1 vor sich.

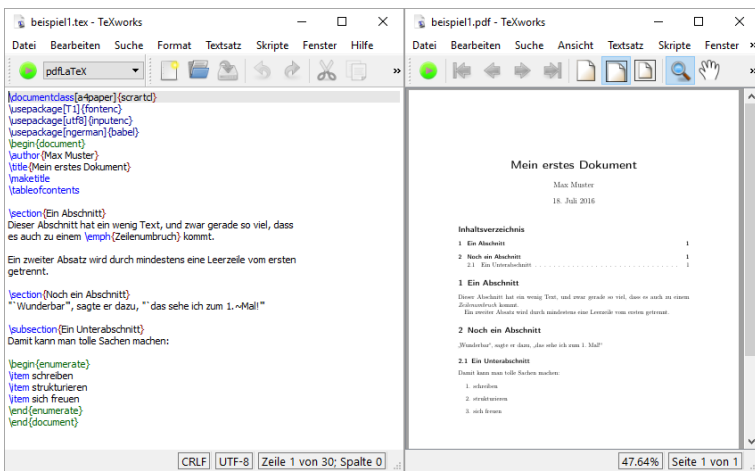


Abb. 3.1: TeXworks-Hauptfenster und Ausgabefenster

3.3 Ein Dokument erstellen und übersetzen

Erzeugen Sie mittels der entsprechenden Schaltfläche ein neues Dokument. Geben Sie jetzt den folgenden Quelltext ein. Um Tippfehler zu vermeiden, können Sie ihn auch direkt von der Website <http://www.latexbuch.de/beispiele> herunterladen und in TeXworks kopieren.

```
1 \documentclass{scartcl}
2 \begin{document}
```

```

3 \section{Ein Abschnitt}
4 Das ist mein erstes Dokument.
5 \end{document}

```

Auf den ersten Blick sieht das Dokument wahrscheinlich ein bisschen kryptisch aus. Welchem Zweck die Befehle im Einzelnen dienen, erfahren Sie später in Kapitel 4 auf Seite 35. Speichern Sie das Dokument jetzt erst einmal als `beispiel0.tex` ab.

Nachdem Sie Ihr Dokument nun verfasst und abgespeichert haben, verwenden Sie einen Übersetzer, um daraus eine Ausgabedatei zu erzeugen. Mehrere Ausgabeformate sind möglich: So kann entweder DVI erzeugt werden – ein \LaTeX -eigenes Format –, das dann nach PostScript weiterverarbeitet werden kann, oder aber PDF (Adobes »Portable Document Format«).

Welches Format Sie wählen sollten, hängt zunächst einmal davon ab, was Sie mit dem fertigen Dokument tun möchten. Sie sollten allerdings auch beachten, dass hier zwei verschiedene Übersetzer zum Einsatz kommen (nämlich \LaTeX und pdf\LaTeX), die teilweise unterschiedlich arbeiten und vor allem unterschiedliche Grafikformate verarbeiten, worauf später in Abschnitt 8.1 auf Seite 135 genauer eingegangen wird.

Enthält die Datei einen fehlerhaften Befehl, den der Übersetzer nicht verarbeiten kann, gibt dieser eine Fehlermeldung aus und unterbricht den Übersetzungsvorgang (auch \LaTeX -Lauf genannt). Genauereres hierzu finden Sie in Abschnitt 15.2 auf Seite 291.

Oft ist es notwendig, zwei oder mehr Übersetzungsvorgänge durchzuführen, um Dinge, die \LaTeX nicht gleich beim ersten Mal komplett erledigen kann (z. B. das Indizieren von Überschriften), vollständig abzuarbeiten.

Beim aktuellen Beispiel brauchen Sie das Dokument aber nur einmal zu übersetzen. Im Folgenden wird zunächst erklärt, wie Sie dazu in **TeXworks** vorgehen. Für den Fall, dass Sie einen Editor verwenden, der diese Funktion nicht vorsieht, ist aber auch die alternative Vorgehensweise über die Kommandozeile beschrieben.

TeXworks Wählen Sie in der Buttonleiste den grünen Pfeil neben »pdfLaTeX«, wenn Sie die Ausgabedatei erzeugen und auch gleich ansehen möchten.

Kommandozeile Ganz allgemein, also ohne an einen bestimmten Editor gebunden zu sein, können Sie auch Folgendes tun: Öffnen Sie eine Kommandozeile unter **Windows** mit einem rechten Mausklick auf das **Windows-Logo**, dann auf **EINGABEAUFFORDERUNG**, und wechseln Sie in das Verzeichnis, in dem Sie Ihre Datei gespeichert haben, beispielsweise mit

```
cd c:\Textdateien
```

Unter Linux haben Sie meist auf dem Desktop ein Shell-Icon und wechseln das Verzeichnis mit

```
cd ~/textdateien
```

Geben Sie nun folgenden Befehl ein:

```
pdflatex beispiel0.tex
```

Wollen Sie PostScript über die Kommandozeile erzeugen, so benötigen Sie zwei Schritte: Zunächst geben Sie

```
latex beispiel0.tex
```

ein, um das Zwischenformat DVI zu erzeugen, bevor Sie mit

```
dvips beispiel0.dvi
```

die PostScriptdatei erhalten.

Welchen Weg auch immer Sie beschritten haben, in Ihrem Verzeichnis sollte es jetzt eine Datei namens `beispiel0.pdf` oder `beispiel0.ps` geben. Diese können Sie nun anzeigen lassen.

3.4 Der Betrachter

Die erzeugte Ausgabedatei `beispiel0.ps` bzw. `beispiel0.pdf` können wir mit GSview bzw. Adobe Reader betrachten, je nachdem, welches Ausgabeformat gewählt wurde.

Auch das Zwischenformat der DVI-Dateien kann mittels eines Programms angesehen werden. DVI-Dateien haben den Vorteil, dass sie relativ klein sind und schneller erzeugt werden als die endgültigen PostScript- oder PDF-Dateien. Aber Vorsicht: Der DVI-Betrachter zeigt nur eine *Voransicht*. Will heißen: Zeilenumbrüche usw. sind korrekt, grafische Elemente wie Linien oder Grafiken können jedoch durchaus einmal anders angezeigt werden als in der späteren Ausgabe. Deswegen sollte eine Endkontrolle immer auch mit dem Betrachter für das endgültige Format durchgeführt werden, also mit dem PDF- oder PostScriptbetrachter.

3.5 Das Dokument

Ein weiteres Beispiel eines Dokuments ist nachfolgend zu sehen (mit Zeilennummern zur besseren Lesbarkeit, diese sind natürlich nicht im Dokument enthalten!):

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage[ngerman]{babel}
5 \usepackage{lmodern}
6 \begin{document}
7 \author{Max Muster}
8 \title{Mein erstes Dokument}
9 \maketitle
10 \tableofcontents
11
12 \section{Ein Abschnitt}
13 Dieser Abschnitt hat ein wenig Text, und zwar gerade so viel, dass
14 es auch zu einem \emph{Zeilenumbruch} kommt.
15
16 Ein zweiter Absatz wird durch mindestens eine Leerzeile vom ersten
17 getrennt.
18
19 \section{Noch ein Abschnitt}
20 ""Wunderbar"", sagte er dazu, ""das sehe ich zum 1.~Mal!""
21
22 \subsection{Ein Unterabschnitt}
23 Damit kann man tolle Sachen machen:
24
25 \begin{enumerate}
26 \item schreiben
27 \item strukturieren
28 \item sich freuen
29 \end{enumerate}
30 \end{document}

```

In diesem zweiten Beispiel sind bereits die meisten wesentlichen Grundelemente eines Dokuments zu sehen, die hier kurz erklärt, später aber noch detaillierter abgehandelt werden.

Befehle sind alle Anweisungen, die L^AT_EX verarbeiten soll. Ein Befehl beginnt immer mit einem Backslash-Zeichen (\). Ein Beispiel ist der

`\section`-Befehl in Zeile 11, der dort einen neuen Abschnitt entstehen lässt. Befehle können *obligatorische* und *optionale* Argumente haben. *Obligatorische* Argumente werden in geschweiften Klammern (`{ }`) angegeben und sind für den jeweiligen Befehl unbedingt notwendig, *optionale* Argumente werden in eckigen Klammern (`[]`) angegeben und können wahlweise auch weggelassen werden.

Mehr dazu finden Sie in Kapitel 4 auf Seite 35.

Umgebungen beginnen immer mit `\begin{umgebung}`² und enden mit `\end{umgebung}`. Umgebungen erlauben es, längere Textteile mit bestimmten Eigenschaften zu versehen. Ein Beispiel dafür ist die `enumerate`-Umgebung in den Zeilen 24 bis 28, die eine Aufzählung beschreibt.

Mehr dazu finden Sie ebenfalls in Kapitel 4 auf Seite 35.

Präambel und Textteil bilden die zwei Teile eines L^AT_EX-Dokuments. Als Präambel werden alle Zeilen bis zum `\begin{document}` bezeichnet. Sie legt alle wesentlichen Eigenschaften des Dokuments fest, also das prinzipielle Aussehen, die Dokumentart etc. Danach folgt – in der `document`-Umgebung – der Textteil, in dem das eigentliche Dokument steht.

Präambel In der Präambel wird das Aussehen des Dokuments und seine Art festgelegt. Dazu wählen Sie eine *Dokumentklasse* aus, laden *Zusatzpakete* und verwenden eventuell einige weitere Befehle.

Klasse Üblicherweise in der ersten Zeile der Datei wird die Klasse und damit die Dokumentart angegeben.

Der Befehl `\documentclass[a4paper]{scrartcl}` legt die Klasse `scrartcl` als Dokumentart fest (obligatorisches Argument) und gibt außerdem die Option `a4paper` mit (optionales Argument). Das bedeutet, unser Beispieltext soll das Layout eines Artikels haben und DIN-A4-Format aufweisen.

Neben der Klasse der Artikels stehen Klassen bereit für Bücher, für wissenschaftliche Arbeiten, für Briefe, für Präsentationen und für fast alles, was man sich sonst an Dokumenten vorstellen kann.

Mehr dazu finden Sie in Kapitel 6 auf Seite 89.

²Oft wird in der Informatik in Syntaxbeschreibungen bei variablen Anteilen mit spitzen Klammern gearbeitet, sodass man `<umgebung>` schriebe. Da ich jedoch festgestellt habe, dass dies viele Leser verwirrt, lasse ich die Klammern weg.

Pakete stellen Zusatzfunktionalitäten zur Verfügung. Sie werden mit dem `\usepackage`-Befehl eingebunden. Im Beispiel:

Das `fontenc`-Paket, das mit der Option `T1` aufgerufen wird, aktiviert die europäischen Zeichensätze, was z. B. der Silbentrennung zugutekommt. Das `inputenc`-Paket mit der Option `utf8` erlaubt die direkte Eingabe von Umlauten und anderen Sonderzeichen. Andernfalls müssten wir anstelle von `ä` etwa `\"a` schreiben. Schließlich stellen wir mit `babel` auf die deutsche Sprache `german` um. Damit stehen deutsche Trennmuster zur Verfügung und alle etwaigen Bezeichnungen im ausgegebenen Dokument sind auf Deutsch. Für die neue Rechtschreibung verwenden Sie `ngerman` statt `german`.

Titel ist alles, was zum Titel gehört. Also zum Beispiel der Titel des Dokuments (`\title`) und der Autor (`\author`). Mit dem Befehl `\maketitle` werden diese Informationen entsprechend aufbereitet und gesetzt. Mehr dazu in Abschnitt 6.4 auf Seite 105.

Struktur Wir können das Dokument mit Befehlen wie `\section` oder `\subsection` strukturieren. Die Nummerierung erfolgt automatisch. Ein Inhaltsverzeichnis erzeugen wir daraus mit `\tableofcontents`. Wie das im Einzelnen funktioniert, steht in Abschnitt 4.2.2 auf Seite 40.

Textauszeichnung heißt es, wenn Sie einen Textteil besonders behandeln möchten. Wollen wir etwa ein Wort hervorheben (engl. »emphasize«), so können wir dies mit dem Befehl `\emph{}` erreichen wie in Zeile 13.

Anführungszeichen kommen in vielen Variationen vor; die deutschen Anführungszeichen können mit `"` (Gänsefüßchen Akzent rechts) und `'` (Gänsefüßchen Apostroph) erzeugt werden, die französischen mit `>` und `<`.

Mehr dazu finden Sie in Abschnitt 4.2.4 auf Seite 46.

Leerraum Zwei Absätze werden durch beliebig viele Leerzeilen getrennt. \LaTeX erkennt dann von selbst, dass hier ein neuer Absatz beginnt. Ebenso können zwischen Wörtern beliebig viele Leerzeichen stehen, der Wortabstand wird von \LaTeX selbst berechnet.

Möchten Sie einen bestimmten Wortzwischenraum, so können Sie dies angeben. Das Zeichen `~` in Zeile 19 erzeugt einen sogenannten *geschützten Leerraum*, das heißt, die beiden dadurch verbundenen Wörter stehen immer in einer Zeile. Weitere Möglichkeiten in diesem Umfeld werden später in Abschnitt 4.3 auf Seite 48 erläutert.

Listen sind Umgebungen und heißen etwa `enumerate` für nummerierte Listen und `itemize` für Aufzählungen. Die einzelnen Listenelemente werden mit dem Befehl `\item` eingeleitet. Alles zu Listen lesen Sie in Abschnitt 4.4 auf Seite 54.

Diese grundlegenden Befehle sind in der einen oder anderen Form in wohl jedem Dokument vorhanden. In den folgenden Kapiteln werden diese sukzessive eingeführt. Das fertige Dokument können Sie in Abbildung 3.2 sehen.

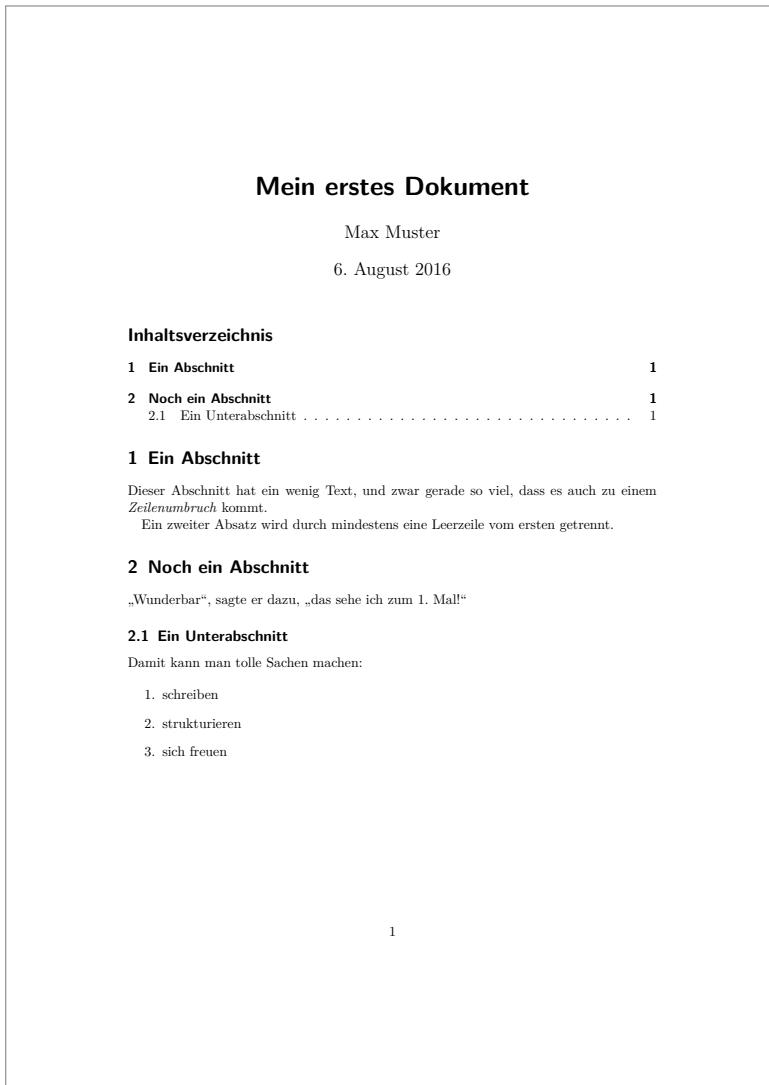


Abb. 3.2: Ausgabe für Beispieldokument 1

Teil II

L^AT_EX-Elemente